

# **JP10309377**

Publication Title:

## **METHOD AND SYSTEM FOR PROVIDING NETWORK INTERFACE**

Abstract:

Abstract of JP 10309377

(A) PROBLEM TO BE SOLVED: To facilitate participation in a game in a multiuser environment by retrieving the list of usable games by capsuling access to a lobby server while using any specified object when scalably connecting a lot of users in the multiuser game. SOLUTION: A data processing system 8 has plural networks such as LAN 10 and 32 and through the LAN 10 and 32, plural individual computers 12 and 30 are connected. Besides, the data processing system 8 has a main frame computer 18 and this is connected to a storage device 20 operable as a remote storage mechanism for the LAN 10. Similarly, the LAN 10 is connected to a subsystem control unit/communication control unit 26 by a communication link 24 and can be connected through a communication link 34 to a gateway server 28. It is preferable that an IWS to be operated for linking the LAN 32 to the LAN 10 is used as the gateway server 28.

---

Courtesy of <http://v3.espacenet.com>

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-309377

(43)公開日 平成10年(1998)11月24日

(51)Int.Cl.<sup>6</sup>  
A 6 3 F 9/22  
G 0 6 F 13/00 3 5 7  
H 0 4 L 12/18  
12/54  
12/58

F I  
A 6 3 F 9/22 C  
G 0 6 F 13/00 3 5 7 Z  
H 0 4 L 11/18  
11/20 1 0 1 B

審査請求 未請求 請求項の数18 O.L (全 12 頁)

(21)出願番号 特願平10-84842

(22)出願日 平成10年(1998) 3月31日

(31)優先権主張番号 08/832430

(32)優先日 1997年4月2日

(33)優先権主張国 米国(US)

(71)出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレイション  
INTERNATIONAL BUSINESS MACHINES CORPORATION  
アメリカ合衆国10504、ニューヨーク州  
アーモンク(番地なし)

(72)発明者 ゲンナロ・エイ・クオモ  
アメリカ合衆国27502 ノースカロライナ  
州アベックス ディア・パレー・ドライブ  
112

(74)代理人 弁理士 坂口 博 (外1名)

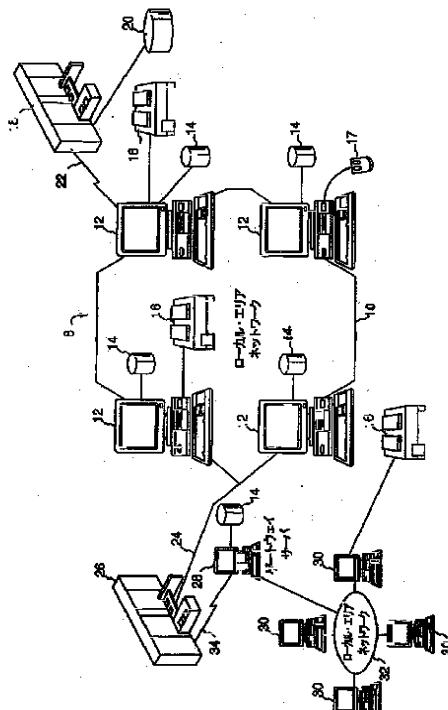
最終頁に続く

(54)【発明の名称】 ネットワーク・インターフェースを提供する方法およびシステム

(57)【要約】

【課題】 マルチユーザ環境におけるゲーム開発者のためのゲーム・ロビーおよびゲーム・セッションを提供すること。

【解決手段】 ゲーム・ロビーにより、ユーザが自分の名前でゲームを中央で開始し、他のユーザがそのゲームを見て参加することができる。ユーザは、このゲームを開始するためにプレーヤの数または開始時間のどちらか最初にくる方などの基準を設定することができる。このパラダイムによって、ゲーム開発者がマルチユーザ通信を管理する負担なしで、プラットフォームに依存しないオブジェクト・モデルが可能になる。このオブジェクト・モデルにより、多数のベンダによるゲーム・ロビー・サービスの開発が可能になり、ゲームをこれらのロビー実施態様のうちのどの実施態様でも自由に操作できるようになる。



**【特許請求の範囲】**

【請求項1】データ処理システムにおいてマルチユーザ・ゲームのために多数のユーザをスケーラブルに接続するためのネットワーク・インターフェースを提供する方法であって、

LobbySessionオブジェクトを使って、ロビー・サーバへのアクセスをカプセル化する段階と、前記LobbySessionオブジェクトを使って、利用可能なゲームのリストを検索する段階とを含む方法。

【請求項2】前記LobbySessionオブジェクトを使って、新しいゲームを作成する段階をさらに含む、請求項1に記載の方法。

【請求項3】前記LobbySessionオブジェクトを使って、既存のゲームに参加する段階をさらに含む、請求項1に記載の方法。

【請求項4】前記LobbySessionオブジェクトを使って、既存のゲームに潜入する段階をさらに含む、請求項1に記載の方法。

【請求項5】前記LobbySessionオブジェクトを使って、GameSessionオブジェクトを獲得する段階をさらに含む、請求項1に記載の方法。

【請求項6】前記GameSessionオブジェクトを使って、GameServerの生成をカプセル化する段階をさらに含む、請求項5に記載の方法。

【請求項7】前記GameSessionオブジェクトを使って、GameServerへのクライアント・アクセスをカプセル化する段階をさらに含む、請求項5に記載の方法。

【請求項8】前記ロビー・サーバがロビー・サーバのクラスタの一部として稼働するように構成されている、請求項1に記載の方法。

【請求項9】前記LobbySessionが、前記クラスタ中の使用可能なロビー・サーバのうちからあるロビー・サーバを選択する、請求項8に記載の方法。

【請求項10】データ処理システムにおいて、マルチユーザ・ゲームのために多数のユーザをスケーラブルに接続するためのネットワーク・インターフェースを提供するためのシステムであって、

LobbySessionオブジェクトを使って、ロビー・サーバへのアクセスをカプセル化する手段と、前記LobbySessionオブジェクトを使って利用可能なゲームのリストを検索する手段とを含むシステム。

【請求項11】前記LobbySessionオブジェクトを使って新しいゲームを作成する手段をさらに含む、請求項10に記載のシステム。

【請求項12】LobbySessionオブジェクトを使って既存のゲームに参加する手段をさらに含む、請求項10に記載のシステム。

【請求項13】前記LobbySessionオブジェクトを使って既存のゲームに潜入する手段をさらに含む、請求項10に記載のシステム。

【請求項14】前記LobbySessionオブジェクトを使ってGameSessionオブジェクトを獲得する手段をさらに含む、請求項10に記載のシステム。

【請求項15】前記GameSessionオブジェクトを使ってGameServerの作成をカプセル化する手段をさらに含む、請求項14に記載のシステム。

【請求項16】前記GameSessionオブジェクトを使ってGameServerへのクライアント・アクセスをカプセル化する手段をさらに含む、請求項14に記載のシステム。

【請求項17】前記ロビー・サーバが、ロビー・サーバのクラスタの一部として稼働するように構成されている、請求項10に記載のシステム。

【請求項18】前記LobbySessionが、前記クラスタ中の利用可能なロビー・サーバのうちからあるロビー・サーバを選択する、請求項17に記載のシステム。

**【発明の詳細な説明】****【0001】**

【発明の属する技術分野】本発明は、一般に、データ処理システムにおいてコンピュータ・ソフトウェアに関し、詳細には、マルチユーザ・ゲームのためにサーバが多数のユーザをスケーラブルに接続できるようにする、通信プロセスへのインターフェースを提供する方法およびシステムに関する。より詳細には、本発明は、ゲーム開発者がユーザを互いに接続し、マルチユーザ・ゲームを作成またはそれに参加することができ、ユーザ間通信をスケーラブルな形で可能にする方法およびシステムに関する。

**【0002】**

【従来の技術】「Quake」（登録商標）、「Populous」（登録商標）、「MUD」（登録商標）（Multiple User Dungeons）などのマルチユーザ・ゲームが増えており、このゲームはクライアント・サーバ・モデルを使った専有物である（proprietary）。このようなゲームの作成者は、通常基本的なゲーム・コンテンツまたはゲーム開発者の専門知識からかなり外れたマルチユーザ通信を管理するタスクに直面している。TEN（Total Entertainment Network）などの会社は、ゲーム作成者にこの通信目標を達成するための高度なプログラミング・インターフェースを提供しようと試みている。このような会社が多数、マルチユーザおよびネットワーク・プログラミングの知識をほとんどまたは全く必要としない簡単なインターフェースを提供するためにしのぎを削っている。

【0003】従来の技術としては、TEN（Total Ente

rтainment Network) やマイクロソフト・コーポレーション (Microsoft Corporation) などの株式未公開会社や公開会社によってインターネット上で提供される専有サービスがある。TENの現在の実施方法は、ゲームを専有ロビー実施態様と統合してユーザにサービスを提供しマルチユーザ・ゲームに参加できるようにするのにゲーム開発者に頼っている。マイクロソフト・コーポレーションは、COMオブジェクトをカプセル化するDirect Playを提供して、そのため開発者はマイクロソフト・プラットフォーム上で実行する必要があり、ActiveXをサポートするブラウザを使用せざるを得ない。ActiveXのモノリシック・アーキテクチャは、複雑なプログラミング・インターフェースが必須である。

【0004】望まれる全体的概念は、ユーザが中央でゲームを自分の名前でオープンすることができ、他のユーザがそのゲームを見て参加することができるゲーム・ロビーを提供することである。ユーザは、プレイヤの数と開始時間のうちどちらか最初にくる方などこのゲームを開始するための基準を設定することができる。本発明は、ゲーム開発者がマルチユーザ通信を管理する負担なしで、プラットフォームに依存しないオブジェクト・モデルによってこのパラダイムを可能にする。このオブジェクト・モデルにより、多数のベンダによるゲーム・ロビー・サービスの開発が可能になり、ゲームをこれらのロビー実施態様のうちのどの実施態様でも自由に操作できるようになる。

#### 【0005】

【発明が解決しようとする課題】本発明は、マルチユーザ環境におけるゲーム開発者のためのゲーム・ロビーおよびゲーム・セッションを提供するための方法およびシステムに関する。

【0006】本発明のもう1つの目的は、どんなマルチユーザ・ゲームをも容易にするためのシームレスなオブジェクト・モデルを提供することである。

【0007】本発明のもう1つの目的は、マルチユーザ環境のための簡単なプログラミング・インターフェースを提供することである。

【0008】本発明のもう1つの目的は、通信システムの使用をいくつかのタイプのネットワークに制限せずにスケーラビリティを提供することである。

【0009】本発明のもう1つの目的は、ゲーム・ロビーを介してゲームを作成しゲームに参加するためのスケーラブルなユーザ接続を最適に容易にするシームレスで構成可能なサーバのクラスタ化を行うことである。

#### 【0010】

【課題を解決するための手段】本発明によれば、前述の目的を達成するために、ここで広義に記述する本発明の方法が提供される。以上その他の本発明の特徴、態様および利点は、以下の説明、併記の特許請求の範囲および添付図面を参照することによりさらによく理解されよ

う。

#### 【0011】

【発明の実施の形態】図1を参照すると、本発明を実施するために利用できるデータ処理システム8の図が示されている。図から分かるように、データ処理システム8は、ローカル・エリア・ネットワーク(LAN)10、32など複数のネットワークを含むことができ、それぞれのネットワークは、複数の個別のコンピュータ12および30を含むことが好ましい。当然ながら、そのような各ネットワークごとに、ホスト・プロセッサに結合された複数のインテリジェント・ワークステーション(IWS)が利用できることを、当業者なら理解するであろう。前記ネットワークはそれぞれ、共用メモリ、共用記憶装置、相互接続ネットワークなどの通信媒体を介して結合された複数のプロセッサを含むことができる。このようなデータ処理システムでは普通のことであるが、個々のコンピュータはそれぞれ、記憶装置14またはプリンタ／出力装置16あるいはその両方に結合することができ、マウス17などのポインティング装置を備えることもできる。

【0012】またデータ処理システム8は、メインフレーム・コンピュータ18など複数のメインフレーム・コンピュータを含むことができ、メインフレーム・コンピュータは、通信リンク22によってLAN10に結合されていることが好ましい。またメインフレーム・コンピュータ18は、LAN10のリモート記憶機構として働くことができる記憶装置20に結合することもできる。同様に、LAN10は、通信リンク24によってサブシステム制御装置／通信制御装置26に結合され、また通信リンク34を介してゲートウェイ・サーバ28に結合されることがある。ゲートウェイ・サーバ28は、LAN32をLAN10にリンクする働きをするIWSであることが好ましい。

【0013】LAN32およびLAN10に関して、複数の文書またはリソース・オブジェクトを記憶装置20内に記憶し、そのように記憶されたリソース・オブジェクトのリソース・マネージャまたはライブラリ・サービスとしてメインフレーム・コンピュータ18によって制御することができる。当然ながら、メインフレーム・コンピュータ18がLAN10から遠く離れた位置に配置され、同様にLAN10がLAN32から十分に遠い距離に配置されることもあることを当業者なら理解されよう。たとえば、LAN32がカリフォルニア州に配置され、LAN10がノースカロライナ州内に配置され、メインフレーム・コンピュータ18がニューヨーク州に配置されていてもよい。

【0014】本発明を利用するソフトウェア・プログラム・コードは、通常、開発者が配布用のコードにそこからアクセスすることができる、独立型ワークステーションまたはLANサーバの記憶装置14のメモリに記憶さ

れ、このソフトウェア・プログラム・コードは、データ処理システムで使用されるディスクケットやCD-ROMなどの様々な既知の媒体のどれでも実施することができ、あるいはあるタイプのネットワークを介してあるコンピュータ・システムのメモリから他のコンピュータ・システムに、そのような他のシステムのユーザが使用できるように配布することができる。媒体上でソフトウェア・コードを実施したまはソフトウェア・コードを配布あるいはその両方を行うそのような技術および方法は周知であり、本明細書ではこれ以上詳しく考察しない。

【0015】図2を参照すると、マルチユーザ・ゲーム環境が示されている。ユーザ200によってマルチユーザ・ゲームが開設されている。ゲームは、ロビー202に登録される。他のユーザは、GameServer204に接続することによりゲームに参加する。現在、さらに2人のユーザ206および208が、GameServer204を介して、ユーザ200が創出したゲームに参加している。新しいユーザ210が、線212で示すようにLobbyServer202と連絡をとつてどのゲームが利用できるかを調べ、当該のGameServersのネットワーク・アドレスを検索し、その後、ユーザは、GameServer204への点線の接続214で示すように、所望のゲームを選択し参加することができる。

【0016】図3は、本発明によるゲーム開発者用のインターフェース・サポート・オブジェクトを示す。オブジェクトは、LobbySession300、GameSession302、GameListener304、およびGameDescriptor306である。この4つのオブジェクトは、ゲームを創出し、創出したゲームをロビーにポストし、ユーザがゲームに参加できるようにする基礎的プロセスをカプセル化したものである。ロビーを介したマルチユーザ・ゲームおよびエントリを容易にするためにはこの4つのオブジェクトだけが必要であり、したがってこれだけがゲーム開発者に開示されている。これは、従来技術で必要とされた比較的大きなAPIとは対照的である。

【0017】図4を参照すると、ゲーム・アプリケーション401は、LobbySession400を使って、参加することができるゲームおよび創出されるゲームを管理するLobbyServer402と通信する。GameDescriptor(GD)404は、ゲーム情報の配布を均一にするためにすべてのオブジェクトによって使用される。LobbySession400から、GameSessionオブジェクト406を作成してGameアプリケーション401に戻し、特定のゲームのプレーヤ間の通信を容易にすることができます。このGameSessionオブジェクト406は、ゲームを創出しゲームに参加するためのサポートを可能にする不透明なデータ・モデルを有する。ゲームが

創出されると、GameSessionオブジェクト406はGameServer410を作成してゲームを管理し、またゲーム作成者の代わりにゲームに参加するためにClientSession408を作成してそのGameServerに接続する。ゲームに参加したとき、GameSessionオブジェクト406は、参加ユーザの代わりに既存の（通常はリモートの）GameServer414に接続するClientSession412を作成するだけである。すなわち、ゲームに参加することによりClientSession408が作成され、ゲームを創出することによりClientSession408とGameServer410が作成される。ゲーム中、アプリケーション401は、次にGameSession406のメソッドを呼び出すことによってゲームを制御することができる。GameSession406は、GameListenerオブジェクト416を介してゲーム・アプリケーション401から供給されるゲーム・イベントに関する通知を受け取る。

【0018】図5を参照すると、本発明のフローチャートが示されている。マルチユーザ・ゲーム用のロビー環境をゲーム開発者がサポートするために、ブロック500で、LobbySessionオブジェクトが作成される。ゲーム開発者は、ロビー・サーバのネットワーク・アドレスまたは位置を指定しなければならない（このロビー・サーバのネットワーク・アドレスは、様々な周知の方法によって得ることができるのでここでは考察しない）。LobbySessionは、ブロック502で、認証のための登録メソッドをサーバに提供する。図5には示していないが、そのユーザは適正な認証がないとロビー・サービスを利用できない場合があることが理解されたい。ブロック504で、使い易くするためにグラフィカル・ユーザ・インターフェース(GUI)がユーザに提供される。次に、判断ブロック510で、ユーザが利用可能なゲームのリストを取得したいと要求したかどうか判定する。判断ブロック510に対する応答がYesの場合、本発明では、ブロック512で、LobbySessionメソッド「getGames」(図3を参照)を使ってゲームへのペクトルを獲得する。次に、本発明は、ブロック504のGUIに戻る。

【0019】判断ブロック510に対する応答がNoの場合は、判断ブロック520で、ユーザがゲームを創出したいかどうか判定する。判断ブロック520に対する応答がYesの場合は、ブロック525で、GameDescriptorオブジェクトがゲーム指定を付けて作成され、LobbySessionの「createGame」メソッド(図3を参照)に渡される。LobbySessionはGameSessionオブジェクトを戻し、ブロック500で以前に作成されたLobbySessionオブジェクトを破壊する。次に、本

発明は600で終了する。

【0020】判断ブロック520に対する応答がNoの場合は、判断ブロック530で、ユーザがゲームに参加したいかどうか判定する。判断ブロック530に対する応答がYesの場合は、ブロック535で、Game DescriptorオブジェクトがGame名の指定付きで作成され、LobbySessionの「join Game」メソッド(図3を参照)に渡される。次に、ブロック537で、LobbySessionはGameSessionオブジェクトを戻し、ブロック500で、以前に作成されたLobbySessionオブジェクトを破壊する。次に、本発明は600で終了する。

【0021】判断ブロック530に対する応答がNoの場合は、判断ブロック540で、ユーザがゲームに「潜入」したい(ユーザはゲームに参加したくないがゲームをするために現在誰が登録されているか確かめたいという意味)かどうか判定する。判断ブロック540に対する応答がYesの場合は、ブロック545で、Game Descriptorオブジェクトが、LobbySession「lurkGame」メソッド(図3を参照)のGame名の指定付きで作成され、指定されたゲームに現在登録されているプレーヤをリストするベクトルを戻す。ブロック545の後、または判断ブロック540に対する応答がNoの場合、本発明は、ブロック504のGU1に戻る。

【0022】図6を参照すると、ロビー・サーバは、全体的に参照番号800で示した複数の独立型サーバとして、あるいは全体的に参照番号810で示しそれぞれサンプル・パラメータの指定802または812による再帰的クラスタ構成で開始することができる。このパラメータ指定の構文は、usageメッセージ814によって記述される。開始されたどのロビー・サーバもLobbySessionが使用することができ、独立型として稼働しようとクラスタの一部として稼働しようと、クラスタ化およびデータ複製がトランスペアレントに実行される。したがって、本明細書で教示する4つのオブジェクト(図3を参照)を使った設計は、独立型構成とは対照的に、プロセスのクラスタ化を利用できる能力を制限しない。

【0023】まとめとして、本発明の構成に関して以下の事項を開示する。

【0024】(1)データ処理システムにおいてマルチユーザ・ゲームのために多数のユーザをスケーラブルに接続するためのネットワーク・インターフェースを提供する方法であって、LobbySessionオブジェクトを使って、ロビー・サーバへのアクセスをカプセル化する段階と、前記LobbySessionオブジェクトを使って、利用可能なゲームのリストを検索する段階とを含む方法。

(2)前記LobbySessionオブジェクトを使

って、新しいゲームを作成する段階をさらに含む、上記(1)に記載の方法。

(3)前記LobbySessionオブジェクトを使って、既存のゲームに参加する段階をさらに含む、上記(1)に記載の方法。

(4)前記LobbySessionオブジェクトを使って、既存のゲームに潜入する段階をさらに含む、上記(1)に記載の方法。

(5)前記LobbySessionオブジェクトを使って、GameSessionオブジェクトを獲得する段階をさらに含む、上記(1)に記載の方法。

(6)前記GameSessionオブジェクトを使って、GameServerの生成をカプセル化する段階をさらに含む、上記(5)に記載の方法。

(7)前記GameSessionオブジェクトを使って、GameServerへのクライアント・アクセスをカプセル化する段階をさらに含む、上記(5)に記載の方法。

(8)前記ロビー・サーバがロビー・サーバのクラスタの一部として稼働するように構成されている、上記(1)に記載の方法。

(9)前記LobbySessionが、前記クラスタ中の使用可能なロビー・サーバのうちからあるロビー・サーバを選択する、上記(8)に記載の方法。

(10)データ処理システムにおいて、マルチユーザ・ゲームのために多数のユーザをスケーラブルに接続するためのネットワーク・インターフェースを提供するためのシステムであって、LobbySessionオブジェクトを使って、ロビー・サーバへのアクセスをカプセル化する手段と、前記LobbySessionオブジェクトを使って利用可能なゲームのリストを検索する手段とを含むシステム。

(11)前記LobbySessionオブジェクトを使って新しいゲームを作成する手段をさらに含む、上記(10)に記載のシステム。

(12)LobbySessionオブジェクトを使って既存のゲームに参加する手段をさらに含む、上記(10)に記載のシステム。

(13)前記LobbySessionオブジェクトを使って既存のゲームに潜入する手段をさらに含む、上記(10)に記載のシステム。

(14)前記LobbySessionオブジェクトを使ってGameSessionオブジェクトを獲得する手段をさらに含む、上記(10)に記載のシステム。

(15)前記GameSessionオブジェクトを使ってGameServerの作成をカプセル化する手段をさらに含む、上記(14)に記載のシステム。

(16)前記GameSessionオブジェクトを使ってGameServerへのクライアント・アクセスをカプセル化する手段をさらに含む、上記(14)に記

載のシステム。

( 17 ) 前記ロビー・サーバが、ロビー・サーバのクラスタの一部として稼働するように構成されている、上記( 10 )に記載のシステム。

( 18 ) 前記LobbySessionが、前記クラスタ中の利用可能なロビー・サーバのうちからあるロビー・サーバを選択する、上記( 17 )に記載のシステム。

( 19 ) データ処理システムにおいてマルチユーザ・ゲームのために多数のユーザをスケーラブルに接続するためのネットワーク・インターフェースを提供するための、媒体に記憶されたコンピュータ・プログラム製品であって、LobbySessionオブジェクトを使ってロビー・サーバへのアクセスをカプセル化するためのコンピュータ可読手段と、前記LobbySessionオブジェクトを使って利用可能なゲームのリストを検索するためのコンピュータ可読手段とを含むコンピュータ・プログラム製品。

( 20 ) 前記LobbySessionオブジェクトを使って新しいゲームを作成するためのコンピュータ可読手段をさらに含む、上記( 19 )に記載のプログラム製品。

( 21 ) 前記LobbySessionオブジェクトを使って既存のゲームに参加するためのコンピュータ可読手段をさらに含む、上記( 19 )に記載のプログラム製品。

( 22 ) 前記LobbySessionオブジェクトを使って既存のゲームに潜入するためのコンピュータ可読手段をさらに含む、上記( 19 )に記載のプログラム製品。

( 23 ) 前記LobbySessionオブジェクトを使ってGameSessionオブジェクトを獲得するコンピュータ可読手段をさらに含む、上記( 19 )に記載のプログラム製品。

( 24 ) 前記GameSessionオブジェクトを使ってGameServerの生成をカプセル化するためのコンピュータ可読手段をさらに含む、上記( 23 )に記載のプログラム製品。

( 25 ) 前記GameSessionオブジェクトを使

ってGameServerへのクライアント・アクセスをカプセル化するためのコンピュータ可読手段をさらに含む、上記( 23 )に記載のプログラム製品。

( 26 ) 上記ロビー・サーバが、ロビー・サーバのクラスタの一部として稼働するように構成されている、上記( 19 )に記載のプログラム製品。

( 27 ) 前記LobbySessionが、前記クラスタ中の利用可能なロビー・サーバのうちからあるロビー・サーバを選択する、上記( 26 )に記載のプログラム製品。

#### 【図面の簡単な説明】

【図1】本発明の方法およびシステムを実施するために利用できるデータ処理システムの絵画図である。

【図2】マルチユーザ・ゲーム環境を示す図である。

【図3】本発明によりゲームを行うためのインターフェース支援オブジェクトを示す図である。

【図4】本発明によるオブジェクトの基礎的アーキテクチャの例を示す図である。

【図5】本発明によるマルチユーザ・ゲームを容易にするフローチャートである。

【図6】本発明によるロビー・サーバの開始を示すフローチャートである。

#### 【符号の説明】

##### 8 データ処理システム

10 LAN

12 コンピュータ

14 記憶装置

16 プリンタ／出力装置

17 マウス

18 メインフレーム・コンピュータ

20 記憶装置

22 通信リンク

24 通信リンク

26 サブシステム制御ユニット／通信制御装置

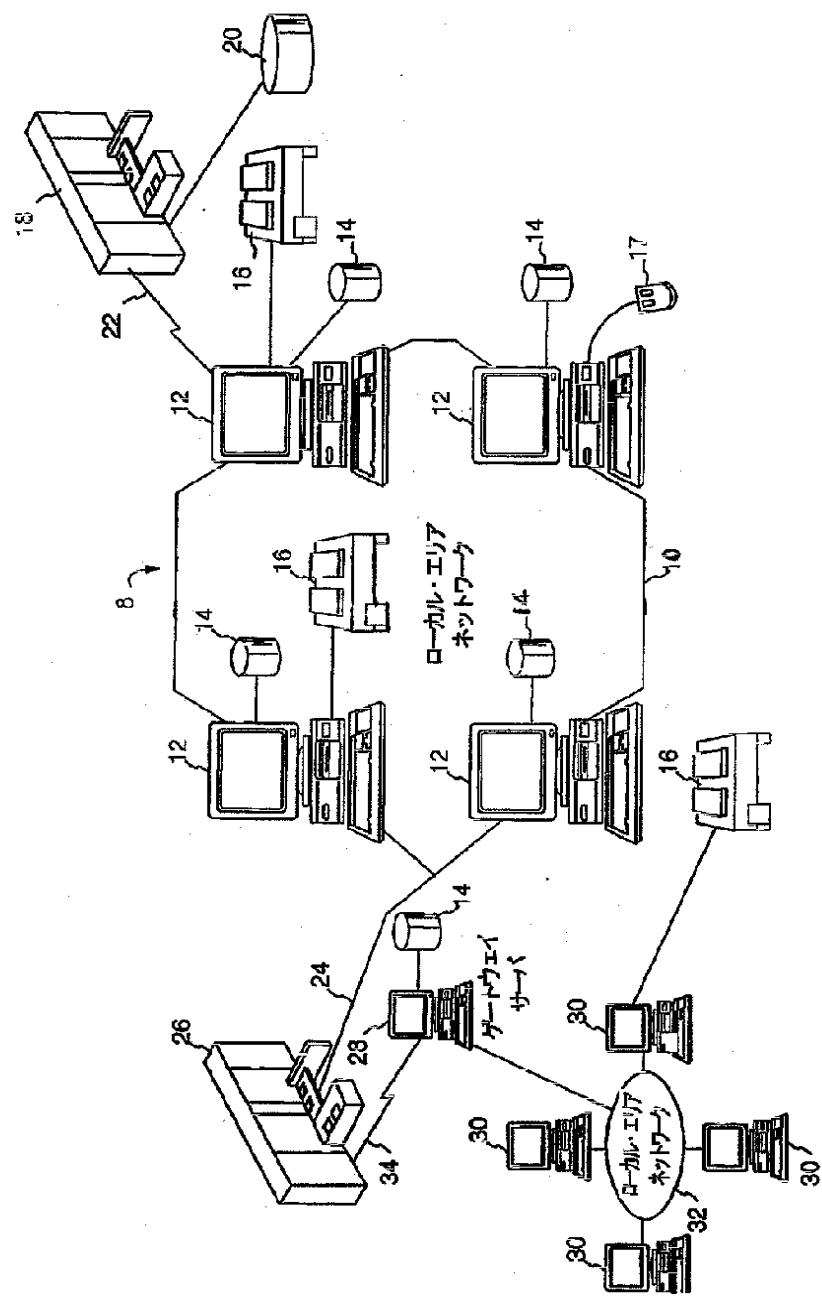
28 ゲートウェイ・サーバ

30 コンピュータ

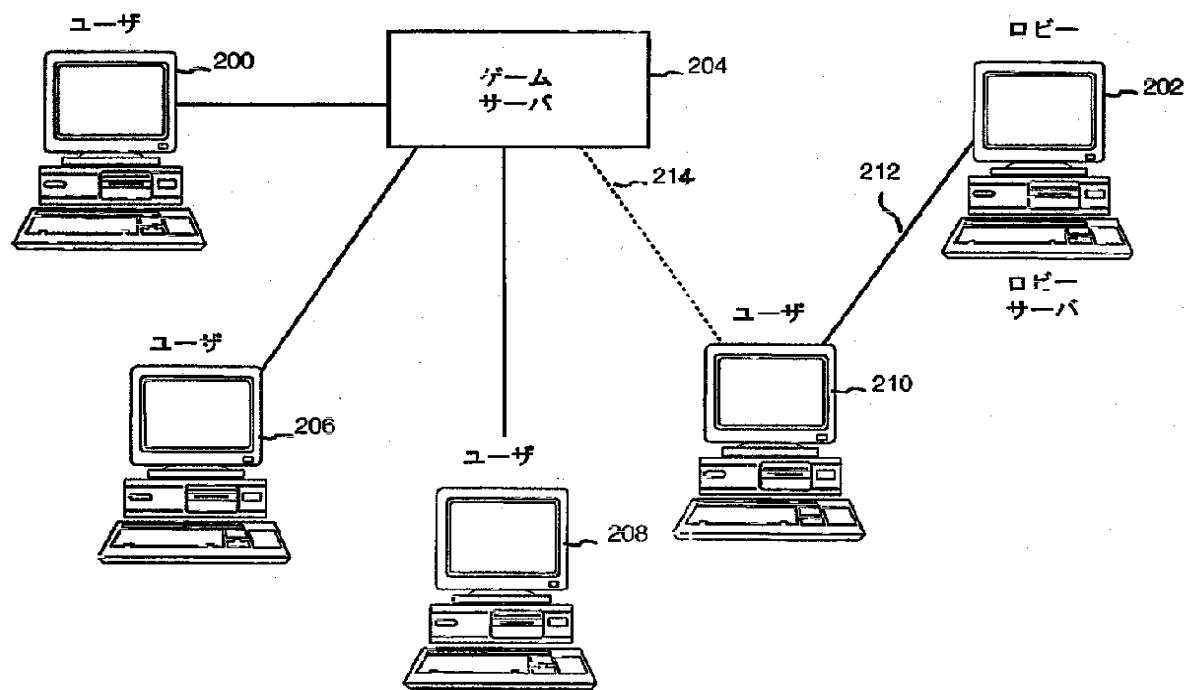
32 LAN

34 通信リンク

【図1】



【図2】



【図3】

```

LobbySession          300
public abstract boolean register(String Password);
public abstract void destroy();
public abstract Vector getGames(String clubs);
public abstract GameSession joinGame(GameDescriptor gd);
public abstract void unjoinGame(GameDescriptor gd);
public abstract Vector LurkGame(GameDescriptor gd);
public abstract GameSession createGame(GameDescriptor gd);
public abstract void set HeartbeatListener(HeartbeatListener hsl);

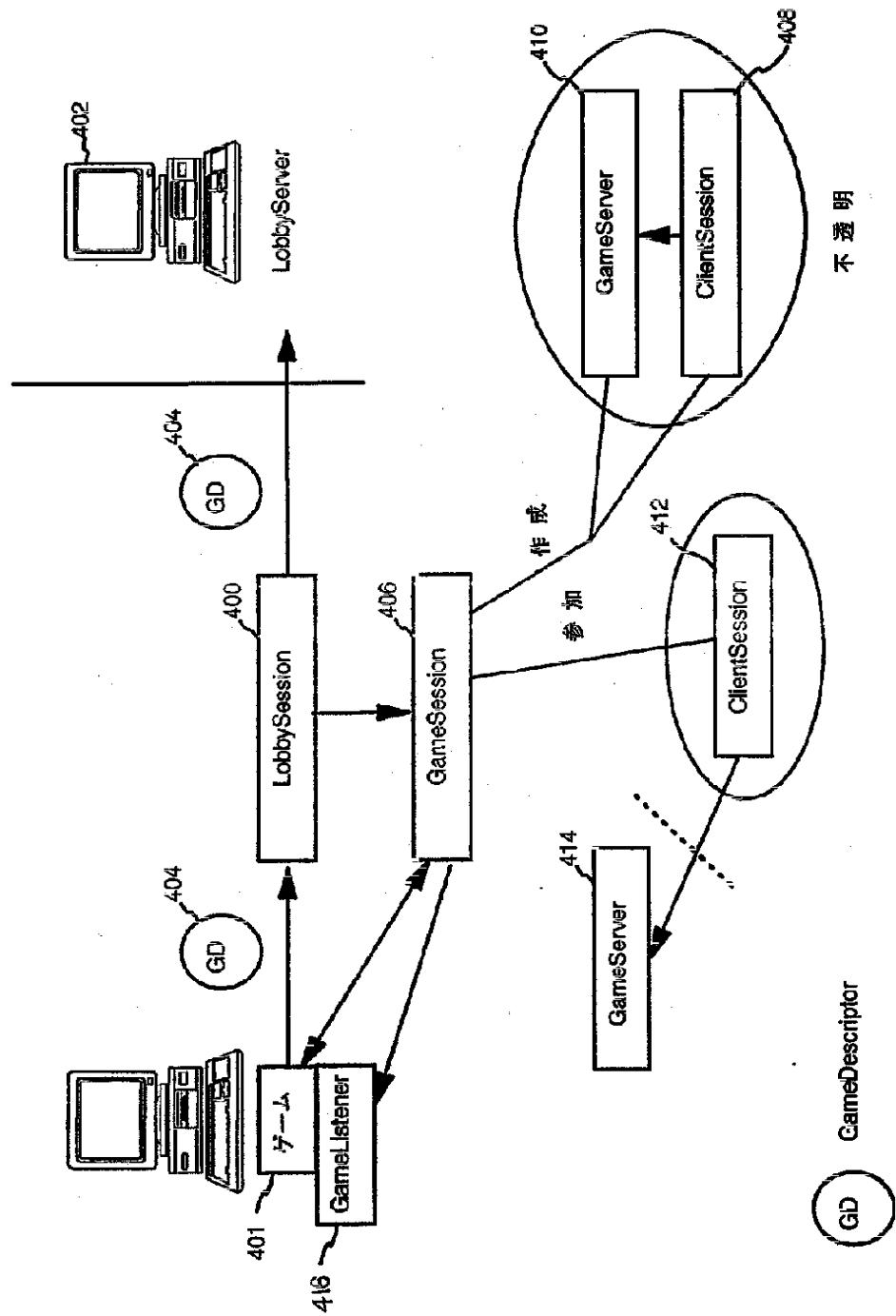
GameSession           302
public abstract void setGameListener(GameListener gl);
public abstract void sendText(String text);
public abstract void sendGameEvent(byte bytes [] );
public abstract void endGame();
public abstract void quitGame();
public abstract void forceStart();
public abstract long currentTIme();

GameDescriptor        306
private String type;
public String universalName;
public String topic;
public String ownerName;
public String ownerHostname;
public String clubs;
public int port;
public int level;
public long startTime;

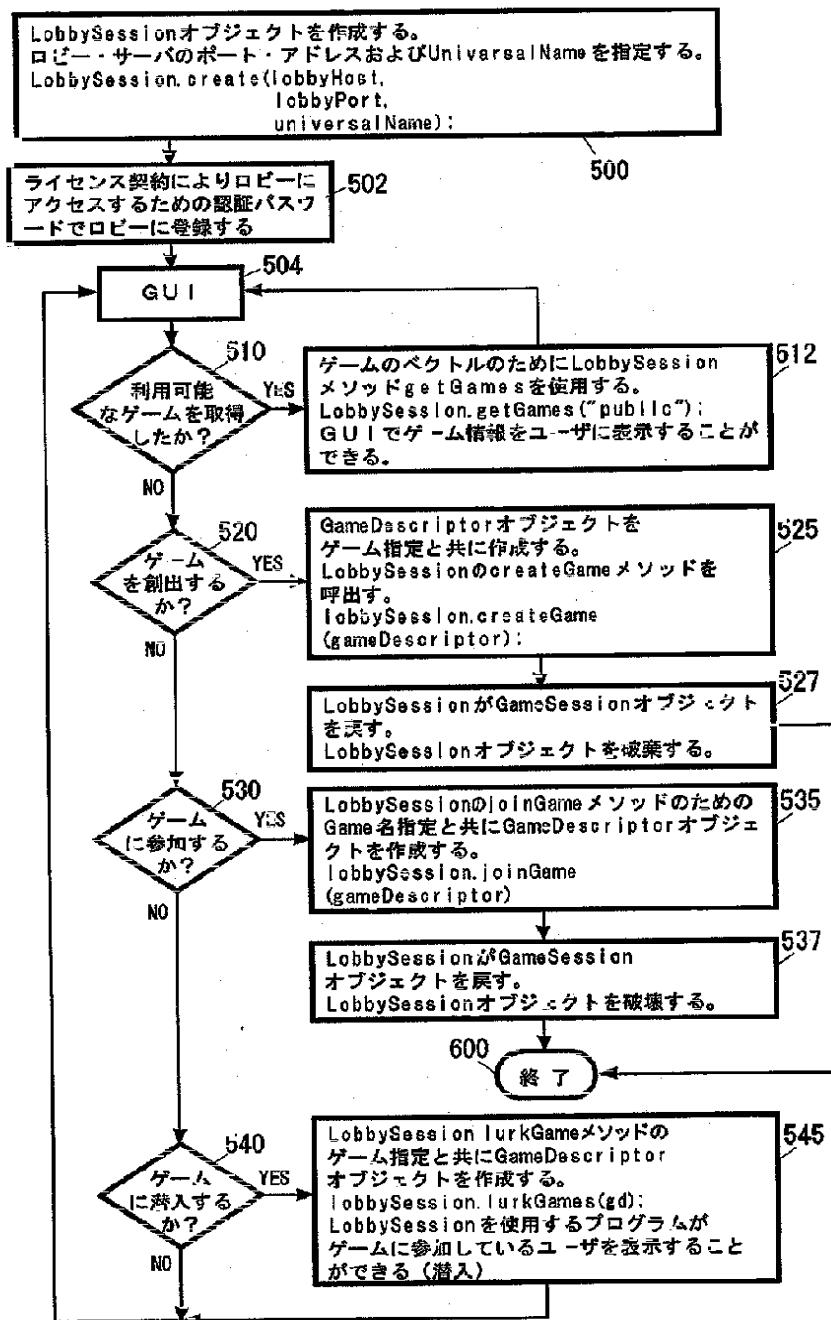
GameListener          304
public abstract void gameStart();
public abstract void gameStop();
public abstract void addPlayer(String universalName, String friendlyName);
public abstract void gameRemovePlayer(String universalName, String friendlyName);
public abstract void gameEvent(Event senderName, byte [] event Parameters);
public abstract void gameNewTextMessage(String SenderuniversalName, String text);

```

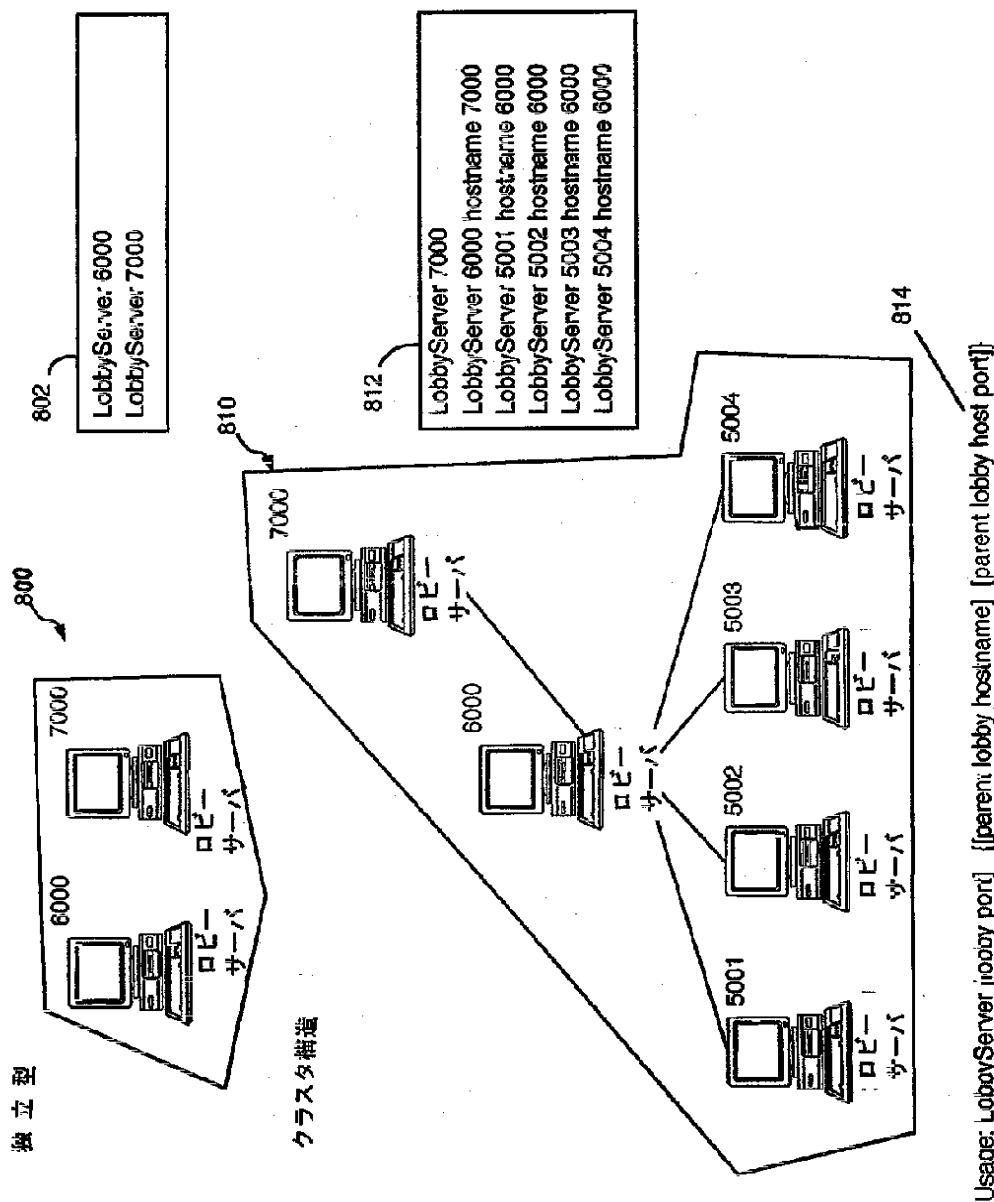
【図4】



【図5】



【図6】



フロントページの続き

(72)発明者 ビン・キュー・グエン

アメリカ合衆国27513 ノースカロライナ  
州ケアリー ベボントン・ドライブ 305

(72)発明者 リチャード・ジェイ・レッドパス

アメリカ合衆国27513 ノースカロライナ  
州ケアリー ブライザウッド・コート  
109

(72)発明者 サンディープ・ケイ・シンハル

アメリカ合衆国27605 ノースカロライナ  
州ローリー ブライトハースト・ドライブ  
1021 ナンバー-105